**Part3**

1. **Write a query to display the current date. Label the column “Date”.**

SELECT SYSDATE AS "Date"

FROM dual

1. **For each employee, display the employee number, last\_name, salary and salary increased by 15% and expressed as a whole number. Label the column New Salary.**

SELECT employee\_id, last\_name, salary, ROUND(salary+salary\*15/100, 0) as "New Salary"

FROM employees

1. **Modify the previous query by adding a column that subtract the old salary from the new salary. Label the column “Increase”.**

SELECT employee\_id, last\_name, salary, ROUND(salary+salary\*15/100, 0) as "New Salary", ROUND(salary+salary\*15/100, 0)-salary as "Increase"

FROM employees

1. **Advanced:** **For each employee display the employee’s last\_name and calculate the number of month between today and the date the employee was hired. Label the column MONTH\_WORKED. Order your results by the number of month employed. Round the number of month up to the closest whole number.**

SELECT last\_name, round(MONTHS\_BETWEEN((Select sysdate from dual), hire\_date),0) MONTH\_WORKED

FROM employees

ORDER BY round(MONTHS\_BETWEEN((Select sysdate from dual), hire\_date),0)

1. **Write a query that produces the following for each employee:**

**<employee last name> earns <salary> monthly but wants <3 times salary>.**

**Label the column “Dream Salaries”.**

SELECT INITCAP(last\_name)||' earns '||salary||' monthly but wants '||salary\*3 "Dream Salaries"

FROM employees

1. **Create a query to display the last name and salary for all employees. Format the salary to be 15 characters long, left-padded with $. Label the column “Salary”.**

SELECT last\_name, LPAD(salary, 15, '$') "salary"

FROM employees

1. **Advanced: Create a query that displays the employees last names and indicates the amounts of their annual salaries with asterisks. Each asterisk signifies a hundred dollars. Sort the data in descending order of salary. Label the column EMPLOYEES\_AND\_THEIR\_SALARIES.**

SELECT last\_name, TRIM('1' FROM LPAD(1,length(round(salary/10,0))+1,'\*')) "EMPLOYEES\_AND\_THEIR\_SALARIES"

FROM employees

1. **Advanced**: **Using the DECODE function, write a query that displays the grade of all employees based on the value of the function JOB\_ID, as per the following data:**

*Job* *Grade*

President A

Salesman B

Clerk C

Manager D

None of above 0

SELECT job\_id,

DECODE (job\_id, 'president', 'A',

'salesman', 'B',

'clerk', 'C',

'manager', 'D', 0) "Grade"

FROM employees

**GROUP FUNCTIONS**

1. **Display the highest, lowest, sum, and average salary of all employees. Label the columns “Maximum”, “Minimum”, “Sum” and “Average”. Round your results to the nearest whole number.**

SELECT MAX(salary) "Maximum", MIN(salary) "Minimum", round(AVG(salary),0) "Average"

FROM employees

1. **Modify the previous query to display the minimum, maximum, sum and average salary for each job type.**

SELECT job\_id, MAX(salary) "Maximum", MIN(salary) "Minimum", round(AVG(salary),0) "Average"

FROM employees

GROUP BY job\_id

1. **Advanced. Write a query to display the number of people with the same job.**

SELECT job\_id, count(\*)

FROM employees

GROUP BY job\_id

1. **Determine the number of managers without listing them. Label the column “Number of Managers”.**

SELECT count(\*) "Number of Managers"

FROM employees

WHERE job\_id='manager'

1. **Write a query that displays the difference between the highest and the lowest salaries. Label the column “Difference”.**

SELECT MAX(salary)-MIN(salary) "Difference"

FROM employees